

MEMORY DEVICE AND METHOD HAVING BANKS OF DIFFERENT SIZES

TECHNICAL FIELD

This invention relates to memory devices, and, more particularly, to a memory device that is adapted for optimal performance to carry out each of a variety of functions.

BACKGROUND OF THE INVENTION

Memory devices, such as dynamic random access memory ("DRAM") devices, include at least one array or bank of memory cells arranged in rows and columns. A row of memory cells is normally accessed by decoding a row address and activating a corresponding word line that extends through the array or bank. When a word line is activated, all of the memory cells in the corresponding row are accessed, and that row is said to be "open." The memory cells in one or more columns of the open row are then accessed by decoding a column address and coupling data bits to or from one or more columns corresponding to the decoded column address.

Opening a row of memory cells normally involves coupling each memory cell in the row to one of a respective pair of complementary digit lines that are provided for each column in the array. Coupling a memory cell to a digit line generates a small differential voltage between the digit lines, which are sensed by a sense amplifiers provided for the column. The amount of power consumed in opening a row is largely proportional to the number of memory cells in the row since power is consumed in opening each memory cell. As a result of the large number of memory cells that are typically in each row, opening a row of memory cells can consume a significant amount of power. However, expending the power to open an entire row provides the advantage of allowing faster access times since data to be read from the memory cells in the open row are available by simply coupling data bits from respective columns. It is very common to sequentially read data from or write data to the memory cells in all or most of the columns of a row. Thus, the need to open an entire row at a time usually does not result in any wasted power. However, there are other memory

operations in which data are read from only a small number of columns in each row. In such cases, opening an entire row, reading data from or writing data to only a few of the memory cells in that row, and then opening a different row needlessly consumes power. For example, if there are 1024 columns in each row, and
5 data bits are read from only 16 of those columns, the amount of power consumed will be approximately 64 times (*i.e.*, 1024/16) the power that would be used to open a row containing only the 16 memory cells that are read.

Not only does opening an entire row to access only a small number of memory cells in the row unnecessarily consume power, but it can also result in slower
10 operation because an array having a large number of columns results in longer word lines having larger capacitances. The larger capacitance of word lines limit the rate at which voltages on the word lines can change to a level that can activate access transistors used to couple respective memory cells to digit lines.

There is therefore a need for a memory device that limits the number of
15 memory cells that are simultaneously opened when only a relatively few memory cells in the open row will be accessed, thereby limiting the needless consumption of power and needless delay in opening rows of memory cells.

SUMMARY OF THE INVENTION

A memory device and method in accordance with the invention includes
20 several arrays or banks of memory cells, at least two of which have different numbers of memory cells in each row. Data are preferably stored in the memory arrays according to the type of data being stored. Data of the type that are normally read from sequential addresses are stored in a memory array having a relatively large number of columns. Data of the type that are normally read from random or non-sequential addresses are
25 stored in a memory array having a relatively small number of columns. The array in which the data are stored can be controlled by a variety of means, including the memory device, a system controller coupled to the memory device or a processor coupled to the system controller that is executing an operating system.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a conceptual block diagram of a memory device according to one embodiment of the invention.

Figure 2 is a block diagram of one embodiment of a memory device
5 based on the concept exemplified by Figure 1.

Figure 3 is a block diagram of a computer system using the memory device of Figure 2.

DETAILED DESCRIPTION OF THE INVENTION

A conceptual block diagram of a memory device 10 according to one
10 embodiment of the invention is shown in Figure 1. The memory device 10 includes 8 arrays or banks of memory cells 14-28, at least some of which are of different sizes. The banks 22, 24 have a larger number of rows and columns than the banks 14-20. The banks 26, 28 have the same number of rows as the banks 14-20 but a larger number of columns than the banks 14-20. For example, the banks 22, 24 may each have 1024
15 rows and 2048 columns of memory cells, the banks 26, 28 may each have 512 rows and 1024 columns of memory cells, and the banks 14-20 may each have 512 rows and 256 columns of memory cells of memory cells.

The banks 14-28 in which the data are stored is preferably a function of their data type. For example, graphics data, which is normally accessed sequentially
20 over a large range of addresses, are stored in the banks 22, 24 having the largest number of columns. Storing graphics data in the large number of columns contained in these banks 22, 24 results in fast access times since data bits are made available from a large number of columns in an open row, but power tends not to be wasted since the data bits stored in all columns of an open row are accessed. Stack data and heap data, which
25 normally consist of a sequence of addresses, are stored in the banks 14, 16 18 and 20, respectively, which have relatively few columns. When accessing heap or stack data, addresses are normally not accessed in sequence, so there is no access time penalty in not opening a row containing a large number of columns. But opening a row containing a large number of columns would tend to needlessly consume power because only a

relatively few columns would be accessed in each open row. Application data, which are normally accessed in sequences that are longer than the sequence of addresses normally used to access heap and stack data but shorter than the sequence of addresses normally used to access graphics data, are stored in the banks 26, 28 having a moderate
5 number of columns. Storing application data in this manner can provide the optimum balance between fast access times and low power consumption.

Although the memory device 10 shown in Figure 1 is preferably a dynamic random access memory (“DRAM”) device, it can alternatively be any type of memory device that stores data in banks of memory cells, including static random
10 access memory (“SRAM”) devices and FLASH memory devices.

A high level block diagram of a dynamic random access memory (“DRAM”) device according to one embodiment of the invention is shown in Figure 2. The DRAM shown in Figure 1 is a synchronous DRAM (“SDRAM”) 100, although, as mentioned above, the invention may be embodied in other types of DRAMs as well as
15 other types of memory devices. The SDRAM 100 includes an address register 104 that receives bank addresses, row addresses and column addresses on a multiplexed address bus. The address bus is generally coupled to a memory controller (not shown) that provides the bank, row and column addresses.

Typically, a bank address and a row address is initially received by the
20 address register 104. The bank address is applied to bank control logic 118, and the row address is applied to a row address multiplexer 108. The bank control logic 118 selects the bank to which data are to be written or from which data are to be read by actuating a row address latch and decoder 120a-d and a column decoder 124 corresponding to a selected bank 130a-d of memory cells. The bank control logic 118 may select one of
25 four memory banks 130a-d to which data are to be written or read based on a variety of factors, such as the nature of the data or the source or destination of the data.

The row address multiplexer 108 couples the row address to a number of components associated with the four memory banks 130a-d depending upon the state of a bank address bit forming part of the row address. Associated with each of the
30 memory banks 130a-d is respective row address latch/decoder 120a-d that stores the

row address and applies various signals to its respective array 130a-d as a function of the stored row address. The row address multiplexer 108 also couples row addresses to the row address latches 120a-d for the purpose of refreshing the memory cells in the arrays 130a-d. The row addresses are generated for refresh purposes by a refresh
5 counter 110. As will be described in greater detail below, the row address multiplexer 108 includes address selection circuitry 109 that selects between providing the external address received from the address register 104 and the address received from the refresh counter 110 to the row address latch/decoder 120a-d as an internal address.

After the row address has been applied to the address register 104 and
10 stored in one of the row address latches/decoder 120a-d, a column address is applied to the address register 104. The address register 104 couples the column address to a column address latch 112 where the address is stored. Column decoders 124a-d receive the column addresses from the column address latch 112, and apply various signals to associated column circuitry 126 and respective sense amplifiers 128a-d for the banks
15 130a-d of memory cells. Data to be read from a selected one of the banks 130a-d are read by actuating a word line in the selected bank, thereby coupling memory cells in the row corresponding to the word line to respective digit lines in the banks 130a-d, one of which is provided for each column of memory cells in the banks 130a-d. Voltages stored in the memory cells are detected by the sense amplifiers 128a-d, one of which is
20 provided for each column of memory cells in each of the banks 130a-d. Data bits corresponding to the detected voltages are then coupled to the column circuitry 126 and subsequently coupled through an internal read path 140 to a data output register 142, which applies the data to a data bus 150. Data to be written to one of the memory banks 130a-d are coupled from the data bus 150 through a data input register 154 and through
25 an internal write path 156 to the column circuitry 126, where the data are transferred to memory cells in the memory banks 130a-d through respective digit lines after actuating a word line for the row of memory cells where the data are to be written.

The SDRAM 100 shown in Figure 2 differs from prior art SDRAM devices by using memory banks 130a-d having differing numbers of columns. For
30 example, BANK0 130a may have 256 columns, BANK1 130b may have 512 columns,

BANK2 130c may have 768 columns, and BANK3 130d may have 1024 columns. As a result, data can be stored in the banks 130a-d depending on the nature of the data. Data that will be sequentially accesses for a considerable number of addresses can be stored in BANK3 130d. On the other hand, data that will be randomly accessed can be stored
 5 in BANK 0 130a to reduce the number of memory cells that are likely to be opened but not subsequently read.

With further reference to Figure 2, the number of sense amplifiers 128a-d used to read data from the memory banks 130a-d varies with the number of columns in the bank 130a-d from which data are being read. When data are being read
 10 from BANK0 130a, the 256 sense amplifiers 128a coupled to that bank are used. When data are being read from BANK1 130b, the 512 sense amplifiers 128b coupled to that bank 130b are used. When data are being read from BANK2 130c, the 768 sense amplifiers 128c coupled to that bank 130c are used. Finally, when data are being read from BANK3 130d, the 1024 sense amplifiers 128d coupled to that bank 130d are used.

15 The above-described operation of the SDRAM 100 is controlled by control logic 102 responsive to high level command signals received on a control bus. These high level command signals, which are typically generated by a memory controller (not shown), often include a clock enable signal CKE*, a clock signal CLK, a chip select signal CS*, a write enable signal WE*, a row address strobe signal RAS*,
 20 and a column address strobe signal CAS*, where the "*" designates the signal as active low. The control logic 102 generates a sequence of command signals responsive to the high level command signals to carry out various memory functions and program memory modes designated by each of the high level command signals, such as memory read, memory write, refresh operations, standby mode, and the like. For example,
 25 driving the RAS* and CAS* inputs low with CKE* high will cause the SDRAM 100 to enter a self-refresh mode. In the self-refresh mode, the control logic 102 causes the memory cells corresponding to the address provided by the refresh counter 110 in the arrays 120a-d to be periodically refreshed. These command signals, and the manner in which they accomplish their respective functions, are conventional. Therefore, in the
 30 interest of brevity, a further explanation of these control signals will be omitted.

In accordance with another aspect of the invention, the control logic 102 includes a mode register 160 that may be programmed by a user. The user may program the mode register 160, for example, to indicate the length of a burst of data that will be accessed in the SDRAM 100 when the SDRAM 100 operates in a burst mode. In burst-

5 mode SDRAMs, the column address identifies a starting address corresponding to the first of a sequence of columns from which data sequentially will be read or to which data subsequently will be written responsive to each cycle of the clock signal CLK applied to the control logic 102. The burst mode programmed in the mode register 160 by the user can also be used to determine the bank 130a-d in which the data are stored.

10 If the mode register 160 is programmed to access data with a long burst, the data will be accessed from the BANK3 130d. If, on the other hand, the mode register is programmed to access data with a short burst, the data will be accessed from either BANK2 130c or BANK1 130b. If the SDRAM 100 is not operating in a burst mode, data may be accessed from BANK0 130a.

15 Although the SDRAM 100 shown in Figure 2 has banks 130a-d that vary by number of columns, it could instead or in addition have banks that vary by number of rows, as in the memory device 10 shown in Figure 1. However, varying the sizes of banks by varying the number of rows does not provide the power saving advantages of varying the sizes of banks by varying the number of columns for the reasons explained

20 above.

Figure 3 shows an embodiment of a computer system 200 that may use the SDRAM 100 or some other memory device according to some other embodiment of the invention. The computer system 200 includes a processor 202 for performing various computing functions, such as executing specific software to perform specific

25 calculations or tasks. The processor 202 includes a processor bus 204 that normally includes an address bus, a control bus, and a data bus. In addition, the computer system 200 includes one or more input devices 212, such as a keyboard or a mouse, coupled to the processor 202 through a system controller 214 to allow an operator to interface with the computer system 200. Typically, the computer system 200 also includes one or

30 more output devices 216 coupled to the processor 202 through the system controller

214, such output devices typically being a printer. One or more data storage devices 218 are also typically coupled to the processor 202 through the system controller 214 to store data or retrieve data from external storage media (not shown). Examples of typical storage devices 218 include hard and floppy disks, tape cassettes, and compact
5 disk read-only memories (CD-ROMs).

The processor 202 is also typically coupled to a cache memory 226, which is usually static random access memory ("SRAM") and to the SDRAM 100 through the system controller 214. The system controller 214 includes an address bus 234 coupled to the address bus (Figure 2) of the SDRAM 100 to couple row addresses
10 and column addresses to the SDRAM 100. The system controller 214 also includes a control bus 238 that couples command signals to the control bus (Figure 2) of the SDRAM 100. The external data bus of the SDRAM 100 is directly coupled to a data bus 240 of the processor 202, although it may alternatively be coupled through the system controller 214. Finally, the computer system 200 may include a graphics
15 processor 260 coupled through the system controller 214 to the processor 202, the SDRAM 100 and the cache memory 226. The graphics processor 260 drives a video monitor 264, such as a cathode ray tube ("CRT") monitor or a liquid crystal display ("LCD") monitor.

According to one embodiment of the invention, the memory controller
20 230 issues a bank address to the SDRAM 100 based on the nature of the data being stored in the SDRAM 100. If the data are the type that will or probably will be accessed sequentially, the memory controller can issue a bank address for BANK3 130d. If the data are the type that will or probably will be accessed randomly, the memory controller can issue a bank address for BANK0 130a.

25 According to another aspect of the invention, the memory controller 230 issues bank addresses based on the device that is requesting access to the SDRAM 100. For example, if the graphics processor 260 is requesting access to the SDRAM 100, the access data are likely to be stored in a long sequence of addresses. Therefore, the system controller 214 will issue a bank address that will cause the data to be stored in
30 the BANK3 130d.

According to still another aspect of the invention, the processor 202 executes an operating system that issues bank addresses to access data in the SDRAM 100 depending on the nature of the data being accessed. For example, when the operating system is accessing graphics data, it will issue a bank address that will cause
5 the data to be stored in the BANK3 130d. On the other hand, when the operating system is accessing heap or stack data, it will issue a bank address that will cause the data to be stored in the BANK0 130a.

From the foregoing it will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration,
10 various modifications may be made without deviating from the spirit and scope of the invention. Accordingly, the invention is not limited except as by the appended claims.